



اَوْنُوْ رَسِيْئِيْ تِيْكَوْ لُوْ كِيْ مَارَا
UNIVERSITI
TEKNOLOGI
MARA

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA,

MERBOK, KEDAH

DIPLOMA IN INFORMATIC LIBRARY

(CDIM144)

PROGRAMMING FOR LIBRARIES

(IML208)

INDIVIDUAL ASSIGNMENT:

PREPARED BY:

FATIN NUR AFIQAH BINTI MUHAMAD FADHIL

CLASS:

KCDIM1443B

PREPARED FOR:

SIR MOHD FIRDAUS MOHD HELMI

SUBMISSION DATE:

18 DECEMBER 2024

INDIVIDUAL ASSIGNMENT:

NAME:

FATIN NUR AFIQAH BINTI MUHAMAD FADHIL

STUDENT ID:

2023687904

CLASS:

KCDIM1443B

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN INFORMATICS LIBRARY

UNIVERSITI TEKNOLOGI MARA,

MERBOK, KEDAH

ACKNOWLEDGEMENT

Alhamdulillah...

First of all, I would like to thank Allah because I could complete the assignment given to me. This assignment has been done with all my effort and hard work to produce a good assignment with full commitment and responsibility.

Also, a big thank you to my Programming for Libraries (IML208) lecturer, Sir Mohd Firdaus Mohd Helmi because without his guidance, I couldn't have completed my assignment. He always gives me a guide on how to do this assignment. I want to say thanks to my friends who always helped me with this assignment and supported me to not give up.

Lastly, I want to say thanks and congratulations to myself because I can complete this assignment before the end of the submission date in good condition.



STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

- a. **Cheating:** Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.
- b. **Plagiarism:** Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behavior that a reasonable person would consider as plagiarism.
- c. **Fabrication:** Falsifying data, information, or citations in any academic assessment and evaluation.
- d. **Deception:** Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.
- e. **Furnishing false information:** Providing false information or false representation to any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in any manner, violate this pledge.

Name: FATIN NUR AFIQAH BINTI MUHAMAD FADHIL

Matric Number: 2023687904

Course Code: IML208

Programme code: CDIM144

Faculty / Campus : UiTM SUNGAI PETANI

*Students are required to sign one pledge for each course taken.

Bahagian Pentaksiran dan Penilaian Akademik 2021

Project Name: Medical Appointment Booking System

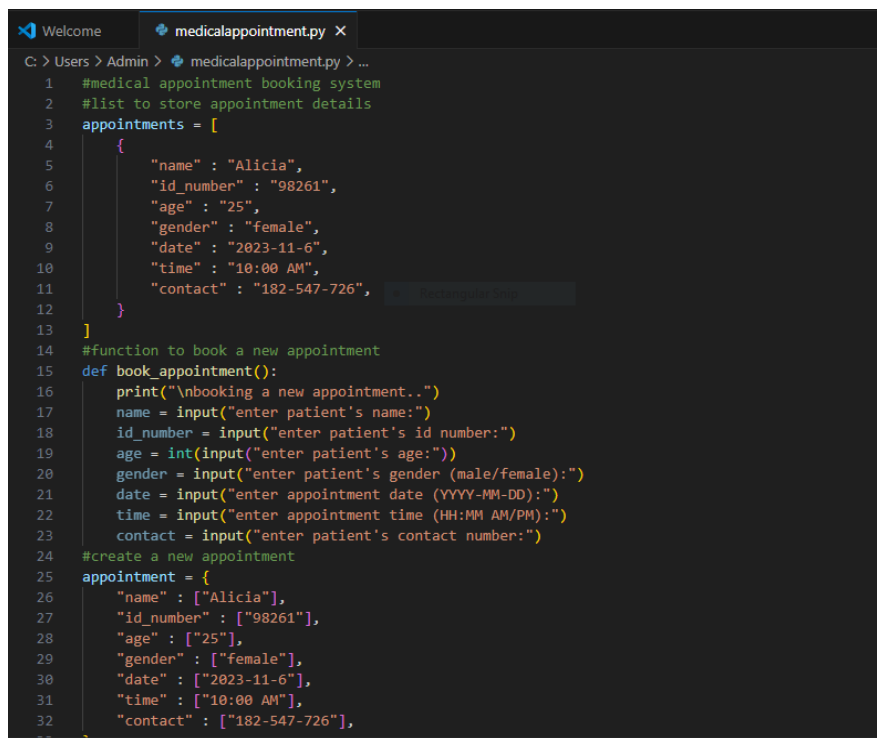
File Name: medical appointment.py

Prompt Data:

- I. **Patient Name:** Alicia
- II. **ID Number:** 98261
- III. **Age:** 25
- IV. **Gender:** Female
- V. **Date Appointment:** 2023-11-6
- VI. **Time Appointment:** 10:00 AM
- VII. **Contact:** 182-547-726

Function:

1. Create Data



```
1 #medical appointment booking system
2 #list to store appointment details
3 appointments = [
4     {
5         "name" : "Alicia",
6         "id_number" : "98261",
7         "age" : "25",
8         "gender" : "female",
9         "date" : "2023-11-6",
10        "time" : "10:00 AM",
11        "contact" : "182-547-726",
12    }
13 ]
14 #function to book a new appointment
15 def book_appointment():
16     print("\nbooking a new appointment..")
17     name = input("enter patient's name:")
18     id_number = input("enter patient's id number:")
19     age = int(input("enter patient's age:"))
20     gender = input("enter patient's gender (male/female):")
21     date = input("enter appointment date (YYYY-MM-DD):")
22     time = input("enter appointment time (HH:MM AM/PM):")
23     contact = input("enter patient's contact number:")
24     #create a new appointment
25     appointment = {
26         "name" : ["Alicia"],
27         "id_number" : ["98261"],
28         "age" : ["25"],
29         "gender" : ["female"],
30         "date" : ["2023-11-6"],
31         "time" : ["10:00 AM"],
32         "contact" : ["182-547-726"],
```

Figure 1: Create patient appointment data

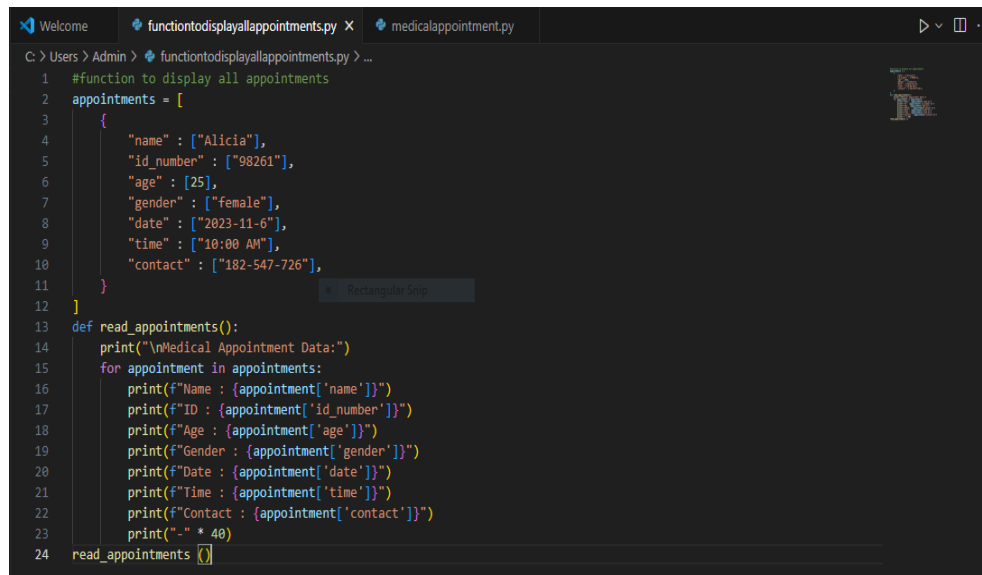
```

33 }
34 appointments.append(appointment)
35 print(f"appointment successfully booked for {\"Alicia\"} on {\"2023-11-6\"} at {\"10:00 AM\"}. ")

```

Figure 2: Create patient appointment data

2. Read Data



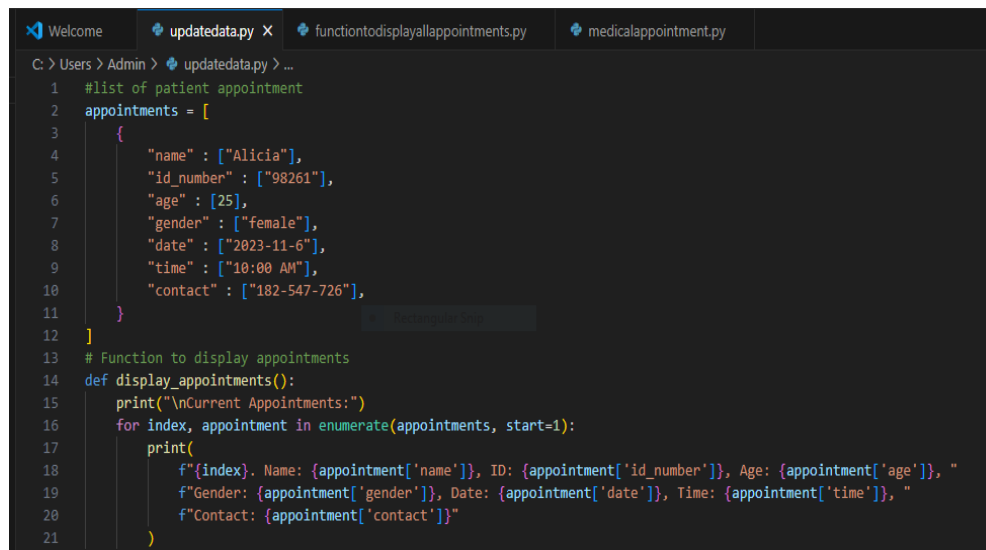
```

Welcome | functiontodisplayallappointments.py | medicalappointment.py
C:\Users\Admin> functiontodisplayallappointments.py
1 #function to display all appointments
2 appointments = [
3     {
4         "name" : ["Alicia"],
5         "id_number" : ["98261"],
6         "age" : [25],
7         "gender" : ["female"],
8         "date" : ["2023-11-6"],
9         "time" : ["10:00 AM"],
10        "contact" : ["182-547-726"],
11    }
12 ]
13 def read_appointments():
14     print("\nMedical Appointment Data:")
15     for appointment in appointments:
16         print(f"Name : {appointment['name']}")
17         print(f"ID : {appointment['id_number']}")
18         print(f"Age : {appointment['age']}")
19         print(f"Gender : {appointment['gender']}")
20         print(f"Date : {appointment['date']}")
21         print(f"Time : {appointment['time']}")
22         print(f"Contact : {appointment['contact']}")
23         print("-" * 40)
24 read_appointments()

```

Figure 3: Read appointment data

3. Update Data



```

Welcome | updatedata.py | functiontodisplayallappointments.py | medicalappointment.py
C:\Users\Admin> updatedata.py
1 #list of patient appointment
2 appointments = [
3     {
4         "name" : ["Alicia"],
5         "id_number" : ["98261"],
6         "age" : [25],
7         "gender" : ["female"],
8         "date" : ["2023-11-6"],
9         "time" : ["10:00 AM"],
10        "contact" : ["182-547-726"],
11    }
12 ]
13 # Function to display appointments
14 def display_appointments():
15     print("\nCurrent Appointments:")
16     for index, appointment in enumerate(appointments, start=1):
17         print(
18             f"{index}. Name: {appointment['name']}, ID: {appointment['id_number']}, Age: {appointment['age']}, "
19             f"Gender: {appointment['gender']}, Date: {appointment['date']}, Time: {appointment['time']}, "
20             f"Contact: {appointment['contact']}"
21         )

```

Figure 4: Update patient appointment data

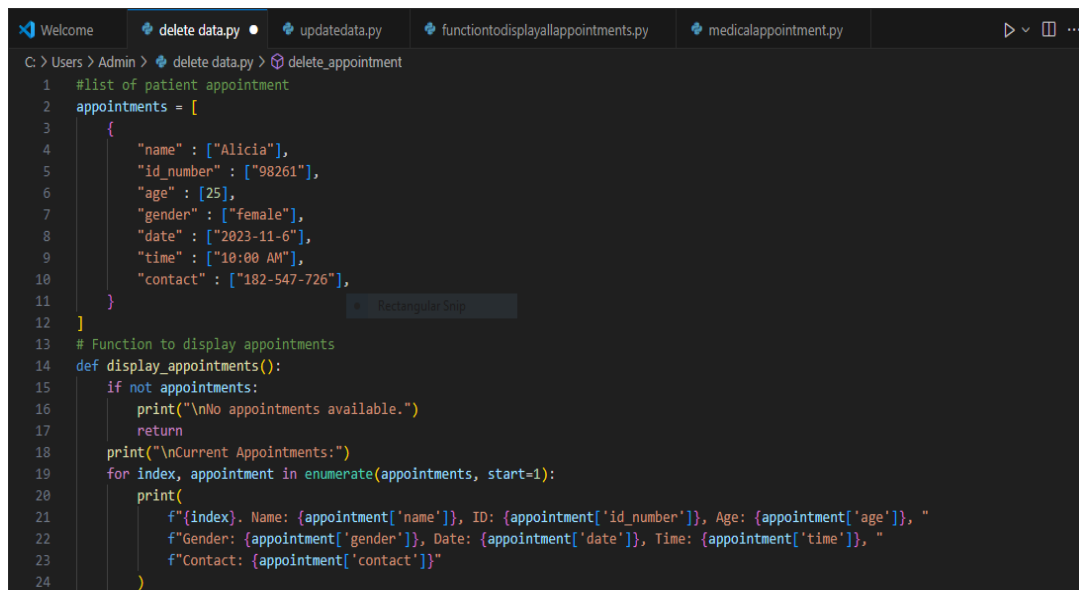
```

22 # Function to update an appointment
23 def update_appointment():
24     display_appointments()
25     try:
26         # Choose an appointment to update
27         choice = int(input("\nEnter the number of the appointment you want to update: ")) - 1
28         if choice < 0 or choice >= len(appointments):
29             print("Invalid choice. Please try again.")
30             return
31         # Select which field to update
32         field = input("Enter the field you want to update (name, id_number, age, gender, date, time, contact): ").strip().lower()
33         if field not in appointments[choice]:
34             print("Invalid field. Please try again.")
35             return
36         # Update the chosen field
37         new_value = input(f"Enter the new value for {field}: ").strip()
38         appointments[choice][field] = new_value
39         print("\nAppointment updated successfully!")
40     except ValueError:
41         print("Invalid input. Please try again.")
42 if __name__ == "__main__":
43     print("Welcome to the Medical Appointment System!")
44     update_appointment()
45     print("\nUpdated Appointments:")
46     display_appointments()

```

Figure 5: Update patient appointment data

4. Delete Existing Data



```

Welcome | delete data.py | updatedata.py | functiontodisplayallappointments.py | medicalappointment.py
C: > Users > Admin > delete data.py > delete_appointment
1 #list of patient appointment
2 appointments = [
3     {
4         "name" : ["Alicia"],
5         "id_number" : ["98261"],
6         "age" : [25],
7         "gender" : ["female"],
8         "date" : ["2023-11-6"],
9         "time" : ["10:00 AM"],
10        "contact" : ["182-547-726"],
11    }
12 ]
13 # Function to display appointments
14 def display_appointments():
15     if not appointments:
16         print("\nNo appointments available.")
17         return
18     print("\nCurrent Appointments:")
19     for index, appointment in enumerate(appointments, start=1):
20         print(
21             f"{index}. Name: {appointment['name']}, ID: {appointment['id_number']}, Age: {appointment['age']}, "
22             f"Gender: {appointment['gender']}, Date: {appointment['date']}, Time: {appointment['time']}, "
23             f"Contact: {appointment['contact']}"
24         )

```

Figure 6: Delete existing patient data

```

25 # Function to delete an appointment
26 def delete_appointment():
27     display_appointments()
28     if not appointments:
29         return
30     try:
31         # Choose an appointment to delete
32         choice = int(input("\nEnter the number of the appointment you want to delete: ")) - 1
33         if choice < 0 or choice >= len(appointments):
34             print("Invalid choice. Please try again.")
35             return
36         # Confirm deletion
37         confirmation = input(f"Are you sure you want to delete the appointment for {appointments[choice]['name']}? (yes/no): ").strip()
38         if confirmation == "yes":
39             deleted_appointment = appointments.pop(choice)
40             print(f"\nAppointment for {deleted_appointment['name']} has been deleted.")
41         else:
42             print("\nDeletion canceled.")
43     except ValueError:
44         print("Invalid input. Please try again.")
45 if __name__ == "__main__":
46     print("Welcome to the Medical Appointment System!")
47     delete_appointment()
48     print("\nUpdated Appointments:")
49     display_appointments()

```

Figure 7: Delete existing patient data

Conditional Statement: Yes

If, elif, & else:

```

Welcome ifelifelse.py x delete data.py updatedata.py functiont
C: > Users > Admin > ifelifelse.py > ...
1 #list of patient appointment
2 appointments = [
3     [
4         "name" : ["Alicia"],
5         "id_number" : ["98261"],
6         "age" : [25],
7         "gender" : ["female"],
8         "date" : ["2023-11-6"],
9         "time" : ["10:00 AM"],
10        "contact" : ["182-547-726"],
11    ]
12 ]
13 #check appointment availability
14 appointment_date = input("enter appointment date:")
15 if appointment_date == "2023-11-6":
16     print("this date already full.")
17 elif appointment_date == "2023-11-10":
18     print("this date is available.")
19 else:
20     print("invalid date or date is not available.")
21
22 appointment_time = input("enter appointment time:")
23 if appointment_time == "10:00 AM":
24     print("this time slot already booked.")
25 elif appointment_time == "3:00 PM":
26     print("this time slot is available.")
27 else:
28     print("invalid time or time is not available.")


```

Figure 8: Check appointment availability

GUI: No

Result:

1. Create Data:

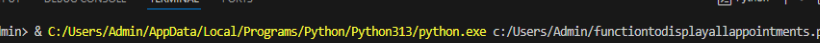


The screenshot shows a Windows command prompt window with the title bar "Python" and standard window controls. The terminal displays the following commands and output:

```
PS C:\Users\Admin> C:\Users\Admin\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Admin/medicalappointment.py
appointment successfully booked for Alicia on 2023-11-6 at 10:00 AM.
PS C:\Users\Admin>
```

Figure 9: Results create a patient appointment data

2. Read Data:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - X
PS C:\Users\Admin> & C:/Users/Admin/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Admin/functiontodisplayallappointments.py

Medical Appointment Data:
Name : ['Alicia']
ID : ['98261']
Age : [25]
Gender : ['female']
Date : ['2023-11-6']
Time : ['10:00 AM']
Contact : ['182-547-726']
-----
PS C:\Users\Admin>
```

Figure 10: Result of read patient appointment data

3. Update Data:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [ ] ... ^ v x

n/Python313/python.exe c:/Users/Admin/contoh.py
Welcome to the Medical Appointment System!

Current Appointments:
1. Name: ['Alicia'], ID: ['98261'], Age: [25], Gender: ['female'], Date: ['2023-11-6'], Time: ['10:00 AM'], Contact: ['182-547-726']

Enter the number of the appointment you want to update: 1
Enter the field you want to update (name, id_number, age, gender, date, time, contact): date
Enter the new value for date: 2023-11-10

Appointment updated successfully!

Updated Appointments:

Current Appointments:
1. Name: ['Alicia'], ID: ['98261'], Age: [25], Gender: ['female'], Date: 2023-11-10, Time: ['10:00 AM'], Contact: ['182-547-726']
PS C:\Users\Admin>

```

Figure 11: Result update patent appointment data

4. Delete Existing Data:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Admin> & C:\Users\Admin\AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Admin/delete_data.py"
Welcome to the Medical Appointment System!

Current Appointments:
1. Name: ['Alicia'], ID: ['98261'], Age: [25], Gender: ['female'], Date: ['2023-11-6'], Time: ['10:00 AM'], Contact: ['182-547-726']

Enter the number of the appointment you want to delete: 1
Are you sure you want to delete the appointment for ['Alicia']? (yes/no): yes

Appointment for ['Alicia'] has been deleted.

Updated Appointments:

No appointments available.
PS C:\Users\Admin>
```

Figure 12: Delete patient appointment data

5. If, Elif, and Else:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Admin> & C:/Users/Admin/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Admin/ifellifelse.py
enter appointment date:2023-11-6
this date already full.
enter appointment time:10:00 AM
this time slot already booked.
PS C:\Users\Admin>

```

Figure 13: Result check appointment availability

Strength:

- Patients can book an appointment anytime online using mobile apps without going to the hospital to make a booking.
- Save time for patients and staff in hospital providers by enabling quick booking, rescheduling, and cancellation.
- Offer convenience for patients to book appointments at their preferred time and date without waiting on hold or visiting the clinic or hospital.
- Patients can see the available slots.
- Flexible to updates like changing the appointment times, and canceling bookings.

Kaizen: (room for improvement)

- Add secure login systems for patients, doctors, and staff
- Allow users to search for appointments by patient name, date, or ID
- Add calendars and forms for booking appointments
- Show available dates and time slots dynamically to avoid conflicts